

# States and channels in quantum mechanics without complex numbers\*

J.A. Miszczak<sup>†</sup>

Institute of Theoretical and Applied Informatics, Polish Academy of Sciences  
Baltycka 5, 44100 Gliwice, Poland

Applied Logic, Philosophy and History of Science group, University of Cagliari  
Via Is Mirrionis 1, 09123 Cagliari, Italy

0.15 (15/03/2016)

## Abstract

In the presented note we aim at exploring the possibility of abandoning complex numbers in the representation of quantum states and operations. We demonstrate a simplified version of quantum mechanics in which the states are represented using real numbers only. The main advantage of this approach is that the simulation of the  $n$ -dimensional quantum system requires  $n^2$  real numbers, in contrast to the standard case where  $n^4$  real numbers are required. The main disadvantage is the lack of hermicity in the representation of quantum states. Using *Mathematica* computer algebra system we develop a set of functions for manipulating real-only quantum states. With the help of this tool we study the properties of the introduced representation and the induced representation of quantum channels.

## 1 Introduction

Quantum information theory aims at harnessing the behavior of quantum mechanical objects to store, transfer and process information. This behavior is, in many cases, very different from the one we observe in the classical world [1]. Quantum algorithms and protocols take advantage of the superposition of states and require the presence of entangled states. Both phenomena arise from the rich structure of the space of quantum states [2]. Hence, to explore the capabilities of quantum information processing, one needs to fully understand this space. Quantum mechanics provides us also with much larger of allowed operations than in classical case space. It can be used to manipulate quantum states. However, the exploration of the space of quantum operations is fascinating, but cumbersome task.

Functional programming is frequently seen as an attractive alternative to the traditional methods used in scientific computing, which are based mainly on the imperative programming paradigm [3]. Among the features of functional languages which make them suitable for the use in this area is the easiness of execution of the functional code in the parallel environments.

During the last few years *Mathematica* computing system has become very popular in the area of quantum information theory and the foundations of quantum mechanics. The main reason for this is its ability to merge the symbolic and numerical capabilities [4], both of which are often necessary to understand the theoretical and practical aspects of quantum systems [5, 6, 7, 8].

---

\*Presented at 21<sup>st</sup> International Conference on Applications of Computer Algebra 2015 (ACA2015), July 20-23, 2015, Kalamata, Greece.

<sup>†</sup>E-mail: miszczak@iitis.pl

In this paper we utilize the ability to merge symbolical and numerical calculations offered by *Mathematica* to investigate the properties of the variant of quantum theory based of the representation of density matrices built using real-numbers only. We start by introducing the said representation, including the *Mathematica* required functions. Next, we test the behavior of selected partial operations in this representation and consider the general case of quantum channels acting on the space of real-only density matrices. In the last part we provide some insight into the spectral properties of the real-only density matrices. Finally, we provide the summary and the concluding remarks.

## 1.1 Preliminaries

In quantum mechanics the state is represented by positive semidefinite, normalized matrix. In the following we focus on this property as it is crucial for the properties of quantum states and channels. To be more specific, we aim at using symbolic matrix which are hermitian. Using the symbolic capabilities of *Mathematica* they can be expressed as

```
SymbolicDensityMatrix[a_, b_, d_] := Array[
  If[#1 < #2, a[#1,#2] + I b[#1,#2],
    If[#1 > #2, a[#2,#1] - I b[#2,#1], a[#1,#2]]] &, {d, d}]
```

In the above definition slots a\_ and b\_ are used to specify the symbols used to denote the real and the imaginary parts of the matrix elements.

Additionally one has to take into account the fact that symbols a\_{i,j} and b\_{i,j} represent real numbers. This fact is useful during the simplifications in the formulas and can be expressed using the function

```
SymbolicDensityMatrixAssume[a_, b_, d_] :=
  $Assumptions = Map[Element[], Reals] &,
  Flatten[Join[
    Table[a_{i,j}, {i,1,d}, {j,1,d}],
    Table[b_{i,j}, {i,1,d}, {j,i+1,d}]
  ]]
]
```

It is easy to see that the normalization condition can be easily added to the list of assumptions. However, the conditions for the positivity, *e.g.* in the form of the positivity conditions for the principal minors, are more complicated [9, Chapter 1].

One should note that, in order to utilize the hermicity conditions for a matrix defined using function **SymbolicDensityMatrix**, is it necessary to execute function specifying assumptions – **SymbolicDensityMatrixAssume** – with the same symbolic arguments.

Another function useful for the purpose of analyzing the operation on quantum states is **SymbolicMatrix** function defined as

```
SymbolicMatrix[a_, d1_, d2_] :=
  Array[Subscript[a, #1, #2] &, {d1, d2}]
```

Using **Flatten** function in combination with **Map** we can impose a list of assumptions on the elements of the symbolic matrix. For example, if one needs to ensure that the elements of the matrix mA are real, this can be achieved as

```
mA = SymbolicMatrix[a, 2, 2];
$Assumptions = Map[Element[], Reals] &, Flatten[mA]]
```

## 2 Using real density matrices

Clearly, the representation of the density used in Section 1.1 is redundant as the off-diagonal element  $a_{i,j} + ib_{i,j}$  is conjugate to  $a_{j,i} - ib_{j,i}$ . Using this observation we can represent any

density matrix as a real matrix with elements defined as

$$\mathcal{R}[\rho]_{ij} = \begin{cases} \text{Re}\rho_{ij} & i \leq j \\ -\text{Im}\rho_{ij} & i > j \end{cases} . \quad (1)$$

The above definition can be translated into *Mathematica* code as

```
ComplexToReal[denMtx_] := Block[{d = Dimensions[denMtx][[1]]},  
  Array[If[#1 <= #2, Re[denMtx[[#1, #2]]],  
    -Im[denMtx[[#1, #2]]] &, {d, d}]  
]
```

Thus, for a given density matrix, describing  $d$ -dimensional system we get a matrix with  $n^2$  real elements, instead of a matrix with  $n^2$  complex (or  $n^4$  real) elements. Note, that these numbers can be reduced during the simulation due to the positivity and normalization conditions, but this requires distinguishing between diagonal and off-diagonal elements.

In the following we denote the map defined by the **ComplexToReal** function as  $\mathcal{R}[\cdot]$ . One should note that  $\mathcal{R} : \mathbb{M}_n(\mathbb{C}) \mapsto \mathbb{M}_n(\mathbb{R})$ . However, we will only consider multiplication by real numbers as it does not affect the hermicity of the density matrix.

The real representation of a density matrix contains the same information as the original matrix. As such it can be used to reconstruct the initial density matrix.

Assuming that `realMtx` represents a real matrix obtained as a representation of the density matrix one can reconstruct the original density matrix as

```
RealToComplex[realMtx_] := Block[{d = Dimensions[realMtx][[1]]},  
  Array[If[#1 < #2, realMtx[[#1, #2]] + I realMtx[[#2, #1]],  
    If[#1 > #2, realMtx[[#2, #1]] - I realMtx[[#1, #2]],  
    realMtx[[#1, #2]]] &, {d, d}]  
]
```

The map defined by the function **RealToComplex** will be denoted as  $\mathcal{C}[\cdot]$ . It is easy to see that for any  $\rho$  we have  $\mathcal{R}[\mathcal{C}[\rho]] = \rho$ .

One can also see that maps  $\mathcal{R}$  and  $\mathcal{C}$  are linear if one considers the multiplication by real numbers only. Thus it can be represented as a matrix on the Hilbert-Schmidt space of density matrices. Using this representation one gets

$$\mathcal{R}[\rho] = \text{res}^{-1} (M_{\mathcal{R}} \text{res}(\rho)) \quad (2)$$

where **res** is the operation of reordering elements of the matrix into a vector [10].

The introduced representation can be utilized to reduce the amount of memory required during the simulation. For the purpose of modelling the discrete time evolution of quantum system, one needs to transform the form of quantum maps into the real representation. For a map  $\Phi$  given as a matrix  $M_{\Phi}$  one obtains its real representation as

$$M_{\mathcal{R}[\Phi]} = M_{\mathcal{R}} M_{\Phi} M_{\mathcal{C}} \quad (3)$$

One can see that this allows the reduction of the number of multiplication operations required to simulate the evolution.

### 3 Examples

Let us now consider some examples utilizing maps  $\mathcal{R}$  and  $\mathcal{C}$ . We will focus on the computation involving symbolic manipulation of states and operations. Only in the last example we use the statistical properties of density matrices which have to be calculated numerically.

#### 3.1 One-qubit case

In the simplest case of two-dimensional quantum system, the symbolic density matrix can be obtained as

**SymbolicDensityMatrix**[a, b, 2]

which results in

$$\begin{pmatrix} a_{1,1} & a_{1,2} + ib_{1,2} \\ a_{1,2} - ib_{1,2} & a_{2,2} \end{pmatrix}. \quad (4)$$

The list of assumptions required to force *Mathematica* to simplify the expressions involving the above matrix can be obtained as

**SymbolicDensityMatrixAssume**[a, b, 2]

which results in storing the following list

$\{a_{1,1} \in \mathbf{Reals}, a_{1,2} \in \mathbf{Reals}, a_{2,2} \in \mathbf{Reals}, b_{1,2} \in \mathbf{Reals}\}$

in the global variable **\$Assumptions**.

In *Mathematica* the application of map  $\mathcal{R}$  on the above matrix results in

$$\begin{pmatrix} \mathbf{Re}(a_{1,1}) & \mathbf{Re}(a_{1,2}) - \mathbf{Im}(b_{1,2}) \\ \mathbf{Re}(b_{1,2}) - \mathbf{Im}(a_{1,2}) & \mathbf{Re}(a_{2,2}) \end{pmatrix}, \quad (5)$$

where **Re** and **Im** are the functions for taking the real and the imaginary parts of the number. Only after using function **FullSimplify** one gets the expected form of the output

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ b_{1,2} & a_{2,2} \end{pmatrix}. \quad (6)$$

In the one-qubit case it is also easy to check that map  $\mathcal{R}$  is represented by the matrix

$$M_{\mathcal{R}}^{(2)} = \frac{1}{2} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -i & i & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}. \quad (7)$$

The matrix representation of the map  $\mathcal{C}$  reads

$$M_{\mathcal{C}}^{(2)} = (M_{\mathcal{R}}^{(2)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & i & 0 \\ 0 & 1 & -i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (8)$$

The above consideration can be repeated and in the case of three-dimensional quantum system the matrix representation of the  $\mathcal{R}$  map reads

$$M_{\mathcal{R}}^{(3)} = \frac{1}{2} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -i & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & -i & 0 & 0 & 0 & i & 0 \\ 0 & 0 & 0 & 0 & 0 & -i & 0 & i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}. \quad (9)$$

### 3.2 One-qubit channels

The main benefit of the real representation of density matrices is the smaller number of multiplications required to describe the evolution of the quantum system.

To illustrate this let us consider a bit-flip channel defined by Kraus operators

$$\left\{ \begin{pmatrix} \sqrt{1-p} & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \begin{pmatrix} 0 & \sqrt{p} \\ \sqrt{p} & 0 \end{pmatrix} \right\}, \quad (10)$$

or equivalently as a matrix

$$M_{BF}^{(2)} = \begin{pmatrix} 1-p & 0 & 0 & p \\ 0 & 1-p & p & 0 \\ 0 & p & 1-p & 0 \\ p & 0 & 0 & 1-p \end{pmatrix}. \quad (11)$$

The form of this channel on the real density matrices is given by

$$M_{\mathcal{R}}^{(2)} M_{BF}^{(2)} M_{\mathcal{C}}^{(2)} = \begin{pmatrix} 1-p & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1-2p & 0 \\ p & 0 & 0 & 1-p \end{pmatrix}. \quad (12)$$

This map acts on the real density matrix as

$$\begin{pmatrix} pa_{2,2} - (p-1)a_{1,1} & a_{1,2} \\ (1-2p)b_{1,2} & pa_{1,1} - (p-1)a_{2,2} \end{pmatrix}. \quad (13)$$

One should note that in *Mathematica* the direct application of the map  $\mathcal{R}$  on the output of the channel, *ie.*  $M_R M_{BF} \text{res } \rho$ , results in

$$\begin{pmatrix} \mathbf{Re}(pa_{2,2} - (p-1)a_{1,1}) & a_{1,2} + 2\mathbf{Im}(p)b_{1,2} \\ (1-2\mathbf{Re}(p))b_{1,2} & \mathbf{Re}(pa_{1,1} - (p-1)a_{2,2}) \end{pmatrix}. \quad (14)$$

In order to get the simplified result one needs to explicitly specify assumptions  $p \in \mathbf{Reals}$ . This is important if one aims at testing the validity of the symbolic computation, as without this assumptions *Mathematica* will not be able to evaluate the result.

### 3.3 Werner states

As the first example of the quantum states of the composite system let us use the Werner states defined for two-qubit systems as

$$W(a) = \begin{pmatrix} \frac{a+1}{4} & 0 & 0 & \frac{a}{2} \\ 0 & \frac{1-a}{4} & 0 & 0 \\ 0 & 0 & \frac{1-a}{4} & 0 \\ \frac{a}{2} & 0 & 0 & \frac{a+1}{4} \end{pmatrix}. \quad (15)$$

The partial transposition transforms  $W(a)$  as

$$W(a)^{TA} = \begin{pmatrix} \frac{a+1}{4} & 0 & 0 & 0 \\ 0 & \frac{1-a}{4} & \frac{a}{2} & 0 \\ 0 & \frac{a}{2} & \frac{1-a}{4} & 0 \\ 0 & 0 & 0 & \frac{a+1}{4} \end{pmatrix}, \quad (16)$$

and this matrix has one negative eigenvalue for  $a > 1/3$ , which indicates a presence of quantum entanglement.

In this case the real representation of quantum states reduces one element from the  $W(a)$  matrix and we get

$$\mathcal{R}[W(a)] = \begin{pmatrix} \frac{a+1}{4} & 0 & 0 & \frac{a}{2} \\ 0 & \frac{1-a}{4} & 0 & 0 \\ 0 & 0 & \frac{1-a}{4} & 0 \\ 0 & 0 & 0 & \frac{a+1}{4} \end{pmatrix}. \quad (17)$$

This matrix has eigenvalues

$$\left\{ \frac{1-a}{4}, \frac{1-a}{4}, \frac{a+1}{4}, \frac{a+1}{4} \right\}, \quad (18)$$

and we have that the sum of smaller eigenvalues is greater than the larger eigenvalue for  $a > 1/3$ .

### 3.4 Partial transposition

Another important example related to the composite quantum systems is the case of partial quantum operations. Such operations arise in the situation when one needs to distinguish between the evolution of the system and the evolution of the same system threatened as a part of a bigger subsystem.

Let us consider the partial transposition of the two-qubit density matrix

$$\rho = \mathbf{SymbolicDensityMatrix}[x, y, 4]$$

which is given by

$$\rho^{T_A} = \begin{pmatrix} x_{1,1} & x_{1,2} + iy_{1,2} & x_{1,3} - iy_{1,3} & x_{2,3} - iy_{2,3} \\ x_{1,2} - iy_{1,2} & x_{2,2} & x_{1,4} - iy_{1,4} & x_{2,4} - iy_{2,4} \\ x_{1,3} + iy_{1,3} & x_{1,4} + iy_{1,4} & x_{3,3} & x_{3,4} + iy_{3,4} \\ x_{2,3} + iy_{2,3} & x_{2,4} + iy_{2,4} & x_{3,4} - iy_{3,4} & x_{4,4} \end{pmatrix}. \quad (19)$$

One can easily check that in this case

$$\mathcal{R}[\rho^{T_A}] = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{2,3} \\ y_{1,2} & x_{2,2} & x_{1,4} & x_{2,4} \\ -y_{1,3} & -y_{1,4} & x_{3,3} & x_{3,4} \\ -y_{2,3} & -y_{2,4} & y_{3,4} & x_{4,4} \end{pmatrix} \quad (20)$$

and

$$(\mathcal{R}[\rho])^{T_A} = \begin{pmatrix} x_{1,1} & x_{1,2} & y_{1,3} & y_{2,3} \\ y_{1,2} & x_{2,2} & y_{1,4} & y_{2,4} \\ x_{1,3} & x_{1,4} & x_{3,3} & x_{3,4} \\ x_{2,3} & x_{2,4} & y_{3,4} & x_{4,4} \end{pmatrix}, \quad (21)$$

and thus

$$\mathcal{R}[\rho^{T_A}] \neq (\mathcal{R}[\rho])^{T_A}. \quad (22)$$

For this reason one cannot change the order of operations. However, the explicit form of the partial transposition on the real density matrices can be found by representing operation of partial transposition as a matrix [10],

**ChannelToMatrix**[**PartialTranspose**[#, {2, 2}, {1}] &, 4]

and using Eq. (3).

One should note that this method can be used to obtain an explicit form of any operation of the form  $\Phi \otimes \mathbb{1}$ , where  $\mathbb{1}$  denotes the identity operation of the subsystem.

### 3.5 Partial trace

The second important example of a partial operation is the partial trace. This operation allows obtaining the state of the subsystem.

For two-qubit density matrix we have

$$\text{tr}_A \rho = \begin{pmatrix} x_{1,1} + x_{3,3} & x_{1,2} + x_{3,4} + i(y_{1,2} + y_{3,4}) \\ x_{1,2} + x_{3,4} - i(y_{1,2} + y_{3,4}) & x_{2,2} + x_{4,4} \end{pmatrix}. \quad (23)$$

One can verify that the operation of tracing-out the subsystem commutes with the map  $\mathcal{R}$  and in this case we have

$$\mathcal{C}[\text{tr}_A \mathcal{R}[\rho]] = \text{tr}_A \rho. \quad (24)$$

Thus one can calculate the reduced state of the subsystem using the real value representation.

### 3.6 Random real states

In this section we focus on the statistical properties of the matrices representing real quantum states. The main difficulty here is that, in contrast to the random density matrices, real representations can have complex eigenvalues.

Random density matrices play an important role in quantum information theory and they are useful in order to obtain information about the average behavior of quantum protocols. Unlike the case of pure states, mixed states can be drawn uniformly using different methods, depending on the used probability measure [2, 11, 12].

One of the methods is motivated by the physical procedure of tracing-out a subsystem. In a general case, one can seek a source of randomness in a given system, by studying the interaction of the  $n$ -dimensional system in question with the environment. In such situation

the random states to model the behaviour of the system should be generated by reducing a pure state in  $N \times K$ -dimensional space. In what follows we denote the resulting probability measure by  $\mu_{N,K}$ .

Using Wolfram language, the procedure for generating random density matrices with  $\mu_{N,K}$  can be implemented as

```
RandomState[n_, k_] := Block{ {gM},
  gM = GinibreMatrix[n, k];
  Chop[#/Tr[#]] &@ (gM. ConjugateTranspose[gM])
}
```

where function **GinibreMatrix** is defined as

```
GinibreMatrix[n_, k_] := Block{ {dist},
  dist = NormalDistribution[0, 1];
  RandomReal[dist, {n, k}] + I RandomReal[dist, {n, k}]
}
```

### 3.7 Spectral properties

In the special case of  $K = N$  we obtain the Hilbert-Schmidt ensemble. The distribution of eigenvalues for  $K = N = 4$  (i.e. Hilbert-Schmidt ensemble for ququart) is presented in Fig. 1.

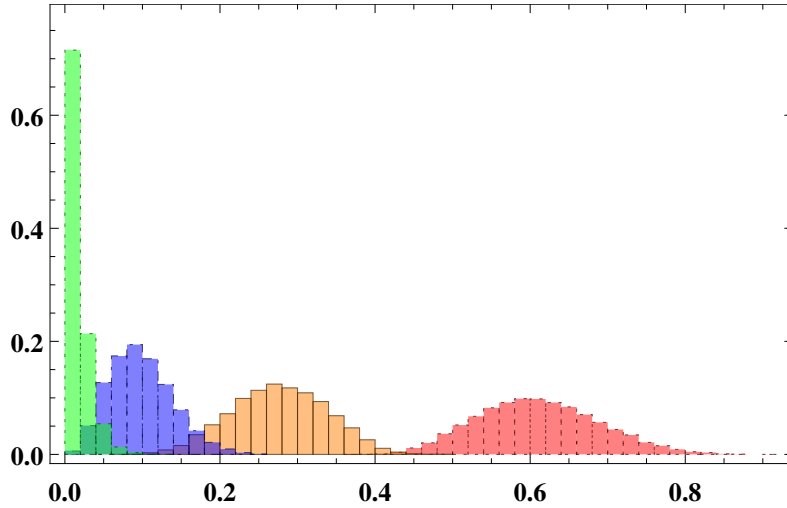


Figure 1: Distribution of eigenvalues for 4-dimensional random density matrices distributed uniformly with Hilbert-Schmidt measure for the sample of size  $10^4$ . Each color (and contour style) correspond to the subsequent eigenvalue, ordered by their magnitude.

The real representation for the Hilbert-Schmidt ensemble for one ququart consists of matrices having four eigenvalues. Two of these values are complex and mutually conjugate (see Fig. 2).

#### 3.7.1 Form of the resulting matrix elements

Using **SymbolicMatrix** function one can easily analyze the dependency of the elements of the resulting matrix on the element of the Ginibre matrix.

For the sake of simplicity we demonstrate this on one-qubit states from the Hilbert-Schmidt ensemble. In this case the Ginibre matrix can be represented as

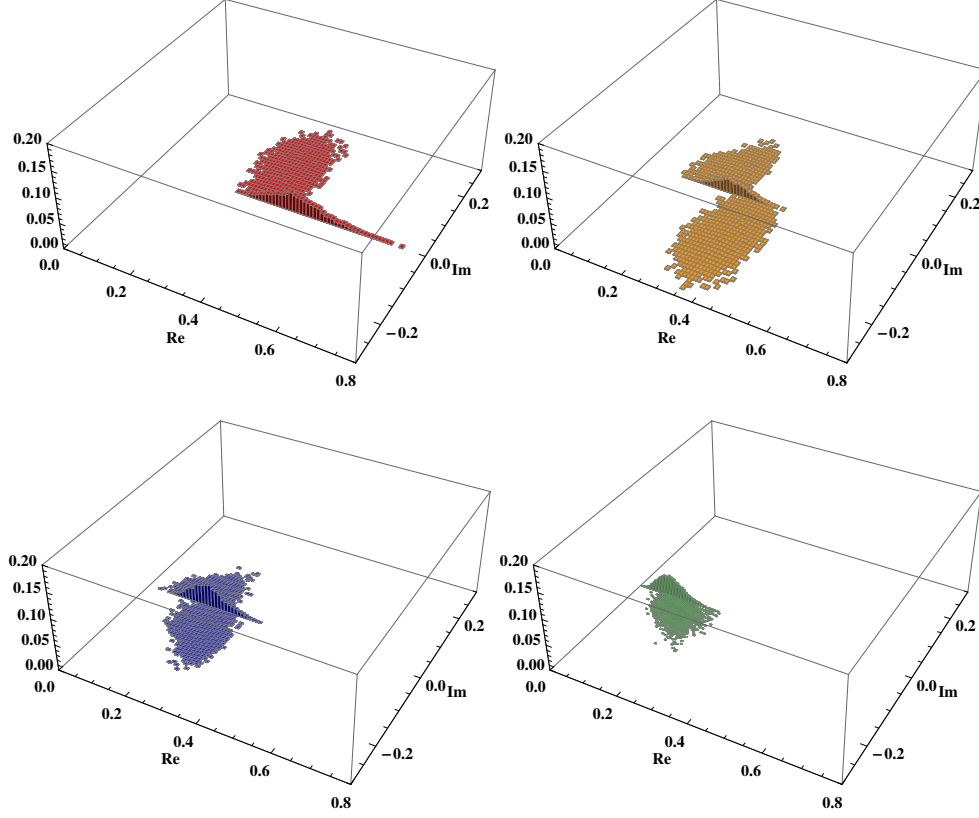


Figure 2: Distribution of eigenvalues for 4-dimensional random density matrices distributed uniformly with Hilbert-Schmidt measure for the sample of size  $10^4$ . Eigenvalues were ordered according to their absolute value.

```

mA = SymbolicMatrix[a, 2, 2];
mB = SymbolicMatrix[b, 2, 2];
m2 = mA + I mB

```

The resulting density matrix has (up to the normalization) elements given by the matrix `m2.ConjugateTranspose[m2]`.

In this case the real representation is given by

$$\begin{pmatrix} q_{1,1} & q_{1,2} \\ q_{2,1} & q_{2,2} \end{pmatrix}, \quad (25)$$

with

$$\begin{aligned} q_{1,1} &= a_{1,1}^2 + a_{1,2}^2 + b_{1,1}^2 + b_{1,2}^2, \\ q_{1,2} &= a_{1,1}a_{2,1} + a_{1,2}a_{2,2} + b_{1,1}b_{2,1} + b_{1,2}b_{2,2}, \\ q_{2,1} &= a_{2,1}b_{1,1} + a_{2,2}b_{1,2} - a_{1,1}b_{2,1} - a_{1,2}b_{2,2}, \\ q_{2,2} &= a_{2,1}^2 + a_{2,2}^2 + b_{2,1}^2 + b_{2,2}^2. \end{aligned} \quad (26)$$

Here  $a_{i,j}$  and  $b_{i,j}$  are independent random variables used in the definition of the Ginibre matrix.

From the above one can see that the elements of the density matrix resulting from the procedure for generating random quantum states are obtained as a product and a sum of the elements of real and imaginary parts of the Ginibre matrix. In the case of density matrices



the normalization imposes the condition  $q_{1,1} = 1 - q_{2,2}$ . Thus, one can also see that the elements are not independent.

## 4 Final remarks

In this note we have introduced a simplified version of quantum states' representation using the redundancy of information in the standard representation of density matrices. Our aim was to find out if such representation can be beneficial from the point of view of the symbolic manipulation of quantum states and operations.

To achieve this goal we have used *Mathematica* computing system to implement the functions required to operate on real quantum states and demonstrated some examples where this representation can be useful from the computational point of view. Its main advantage is that it can be used to reduce the memory requirements for the representation of quantum states. Moreover, in some particular cases where the density matrix contains only real numbers, the real representation reduces to the upper-triangular matrix.

The real representation can be also beneficial for the purpose of modelling quantum channels. Here its main advantage is that it can be used to reduce the number of multiplications required during the simulation of the discrete quantum evolution. As a particular example, we have studied the form of partial quantum operations in the introduced representation. In the case of the partial trace for the bi-bipartite system, the introduced representation allows the calculation of the reduced dynamics using the real representation only.

Unfortunately, the introduced representation poses some disadvantages. The main drawback of the introduced representation is the lack of hermicity of real density matrices. This makes the analysis of the spectral properties of real quantum states much more complicated.

**Acknowledgement** This work has been supported by Polish National Science Centre project number 2011/03/D/ST6/00413 and RAS project on: "Modeling the uncertainty: quantum theory and imaging processing", LR 7/8/2007. The author would like to thank G. Sergioli for motivating discussions.

## References

- [1] J.A. Miszczak. *High-level structures for quantum computing*, volume #6 of *Synthesis Lectures on Quantum Computing*. Morgan and Claypol Publishers, May 2012. DOI: 10.2200/S00422ED1V01Y201205QMC006.
- [2] I. Bengtsson and K. Życzkowski. *Geometry of Quantum States: An Introduction to Quantum Entanglement*. Cambridge University Press, Cambridge, U.K., 2006. DOI: 10.1017/CBO9780511535048.
- [3] K. Hinsén. The promises of functional programming. *Comput. Sci. Eng.*, 11(4):86–90, 2009. DOI: 10.1109/MCSE.2009.129.
- [4] S. Wolfram. *An Elementary Introduction to the Wolfram Language*. Wolfram Media, Inc., 2015. ISBN: 9781944183004.
- [5] B. Juliá-Díaz, J.M. Burdis, and F. Tabakin. QDENSITY—a Mathematica quantum computer simulation. *Comp. Phys. Comm.*, 174:914–934, 2006. DOI: 10.1016/j.cpc.2005.12.021.
- [6] F. Tabakin and B. Juliá-Díaz. QCWAVE – a Mathematica quantum computer simulation update. *Comp. Phys. Comm.*, 182(8):1693 – 1707, 2011. DOI: 10.1016/j.cpc.2011.04.010, arXiv:1101.1785.
- [7] V.P. Gerdt, R. Kragler, and A.N. Prokopenya. A Mathematica package for simulation of quantum computation. In V.P. Gerdt, E.W. Mayr, and E.V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing / CASC2009*, volume 5743 of *LNCs*, pages 106–117. Springer-Verlag, Berlin, 2009. DOI: 10.1007/978-3-642-04103-7\_11.

- [8] J.A. Miszczak. Functional framework for representing and transforming quantum channels. In J.L. Galan Garcia, G. Aguilera Venegas, and P. Rodriguez Cielos, editors, *Proc. Applications of Computer Algebra (ACA2013), Malaga, 2-6 July 2013*, 2013. arXiv:1307.4906.
- [9] R. Bhatia. *Positive Definite Matrices*. Princeton University Press, Princeton, U.S.A., 2007. DOI: 10.1515/9781400827787.
- [10] J.A. Miszczak. Singular value decomposition and matrix reorderings in quantum information theory. *Int. J. Mod. Phys. C*, 22(9):897–918, 2011. DOI: 10.1142/S0129183111016683, arXiv:1011.1585.
- [11] J.A. Miszczak. Generating and using truly random quantum states in Mathematica. *Comput. Phys. Commun.*, 183(1):118–124, 2012. DOI: 10.1016/j.cpc.2011.08.002, arXiv:1102.4598.
- [12] J.A. Miszczak. Employing online quantum random number generators for generating truly random quantum states in mathematica. *Comput. Phys. Commun.*, 184(1):257258, 2013. DOI: 10.1016/j.cpc.2012.08.012, arXiv:1208.3970.